PiLocNet: Physics-informed neural network on 3D localization with rotating point spread function

MINGDA LU,[1] ZITIAN AO,[2] CHAO WANG,[2,3,*] SUDHAKAR PRASAD,[4] AND RAYMOND CHAN[5,*]

[1]*Department of Mathematics, City University of Hong Kong, Hong Kong SAR, China*
[2]*Department of Statistics and Data Science, Southern University of Science and Technology, Shenzhen 518005, Guangdong Province, China*
[3]*National Centre for Applied Mathematics Shenzhen, Shenzhen 518055, Guangdong Province, China*
[4]*School of Physics and Astronomy, University of Minnesota, USA*
[5]*Lingnan University, Hong Kong SAR, China*
[*]*wangc6@sustech.edu.cn; raymond.chan@ln.edu.hk*

Abstract: For the 3D localization problem using point spread function (PSF) engineering, we propose a novel enhancement of our previously introduced localization neural network, LocNet. The improved network is a physics-informed neural network (PINN) that we call PiLocNet. Previous works on the localization problem may be categorized separately into model-based optimization and neural network approaches. Our PiLocNet combines the unique strengths of both approaches by incorporating forward-model-based information into the network via a data-fitting loss term that constrains the neural network to yield results that are physically sensible. We additionally incorporate certain regularization terms from the variational method, which further improves the robustness of the network in the presence of image noise, as we show for the Poisson and Gaussian noise models. This framework accords interpretability to the neural network, and the results we obtain show its superiority. Although the paper focuses on the use of a single-lobe rotating PSF to encode the full 3D source location, we expect the method to be widely applicable to other PSFs and imaging problems that are constrained by well modeled forward processes.

1. Introduction

Locating point sources or structures in three-dimensional (3D) space is a common challenge in many scientific applications. This is particularly relevant in computer vision, which has numerous applications like robotics, augmented reality, and autonomous systems. For solving the 3D localization problem, point spread function (PSF) engineering is a promising and effective technique that places a specific phase function into the imaging aperture. This aperture function processes the photons emitted by a source point and entering the imaging system into an image pattern that carries information about the full 3D location of the source. Unlike traditional methods that leverage stereo or multi-view images, PSF engineering imprints the 3D location information of the point sources into the position and form of the corresponding images acquired on a single two-dimensional (2D) sensor. PSF-associated methodologies have wide applications that range from telescopic to microscopic imaging systems and have significantly enhanced the precision of point source localization. For example, single-molecule localization microscopy (SMLM) [1] localizes individual fluorophores in 3D structures to render super-resolution imaging of fluorescent molecules. By leveraging the z-dependent form of the PSFs, 3D SMLM surpasses traditional diffraction limits, allowing for the visualization of biological structures at near-molecular resolution in all three dimensions. Numerous field studies have underscored this

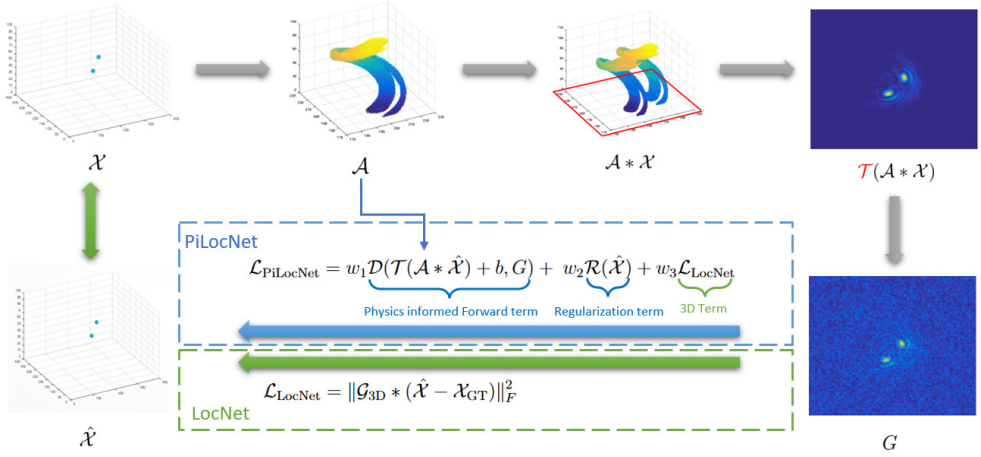


Fig. 1. Overview of the PiLocNet: the main improvement is in the inclusion of a forward loss term based on the matrix $\mathcal{A}$ that models the 3D point spread function along with an appropriate regularization term into the loss function. This added physics information improves network training.

methodology's significance and potential [2–5].

Various depth-encoding phase masks have been developed for PSF engineering, yielding a variety of PSFs, such as astigmatic [6], double helix (DH) [7], and tetrapod [8]. Here, we mainly consider the single-lobe rotating PSF (RPSF) invented by Prasad [9]. By means of a suitable spiral phase function in the imaging aperture, one can create a PSF that rotates by an angle proportional to the depth (z) coordinate of the point source around a center fixed by the 2D transverse (x, y) coordinates of the source. The rotating PSF comprises a single bright lobe surrounded by a fainter ring-shaped substructure, which rotates together as the source defocus distance along the optical (z) axis changes. One of the main benefits of a single-lobe rotating PSF over other more complicated PSFs like the DH and tetrapod PSFs is that the former concentrates the photon energy into its single lobe with a higher flux density, making it more noise-robust in crowded source fields [10].

The main objective of the problem is to recover the 3D locations of point sources from their observed noisy image as accurately as possible. Various methods have been proposed and categorized into mathematical optimization and neural network approaches. Several variational methods [11–14] have been recently introduced from a non-convex optimization perspective. In the case of Gaussian image noise, the Frobenius norm is used as the data fitting term, and a continuous exact ℓ_0 penalty (CEL0) is the regularization term. For the case of Poisson noise, the KL-NC model [14] uses the Kullback-Leibler (KL) divergence data-fitting term and a different non-convex regularization term. The optimization problem is solved by an iteratively reweighted ℓ_1 algorithm.

Deep-learning neural network approaches have also been proposed to solve this 3D localization problem. Two important architectures are DeepSTORM3D [15] and DECODE [16]. Specifically, DeepSTORM3D uses a 3D grid network to map and predict the coordinates of the point sources, while DECODE uses a different structure, with multiple channels, to predict different kinds of information about the input images, including the 3D coordinates, brightness, and the probability of existence of point sources. Recently, Dai et al. proposed LocNet [17], adapting DeepSTORM3D on single-lobe rotating-PSF images, with an additional post-processing step to

cluster the initial prediction of the network.

In the field of neural network studies, the method of physics-informed neural network (PINN) has emerged in recent years [18–21], initially in the context of problems involving partial differential equations (PDEs) but later applied more widely. The goal is to incorporate any known physics information about the problem either directly into the neural network structure or via the loss function.

Inspired by the idea of PINN, we propose here the Physics-Informed Localization Network (PiLocNet) for the PSF localization problem, as shown schematically in Fig. 1. Given the process of PSF image generation, the forward model is a known piece of physical information. Supplying such physical information to the neural network is helpful to the network in generating better results than a random black-box kind of fitting approach characteristic of more conventional neural networks. With model-specific data fitting and regularization terms, a PINN-based method is interpretable to the neural network. The numerical experiments we report here have been conducted based on the RPSF model. Our results, as we will show, prove that the added physical information can significantly improve the prediction accuracy in terms of both precision and recall rates. Our ablation studies also verified the robustness of PiLocNet.

The rest of this paper is organized as follows: In Section 2, we briefly review the optical model of the rotating PSF and the variational methods we used previously for this localization problem. In Section 3, we introduce the specific model structure, including an improved loss function that encompasses both the forward model and regularization terms. Next, we introduce a series of simulation-based experiments that we conducted to verify the effects of our PiLocNet in Section 4. We conclude the paper with a summary of our findings and future work in Section 5.

2. Single-lobe point spread function and its noise models

This section will provide a review of the single-lobe rotating PSF forward model and the two different noise models that we explore in the paper.

2.1. Forward model of single-lobe RPSF

The forward model has been described in great detail in our previous papers [9, 13, 14]. Here, we only present a brief summary of the model. The RPSF image, $\mathcal{A}_\zeta$, for a point source with defocus parameter ζ, a unit flux $f = 1$, at the source location $\mathbf{r}_O = (x_O, y_O)$ is given by:

$$A_\zeta(\mathbf{s}) = \frac{1}{\pi}\left|\int_\Omega \exp\left[i\left(2\pi\mathbf{u}\cdot\mathbf{s} + \zeta u^2 - \psi(\mathbf{u})\right)\right] d\mathbf{u}\right|^2,$$

where $\zeta = -\frac{\pi\delta z R^2}{\lambda z_O(z_O+\delta z)}$, Ω represents the circular disk-shaped clear pupil of radius R, and $i = \sqrt{-1}$. The quantity, $\mathbf{s} = \frac{\mathbf{r}}{\lambda z_I/R}$, is the position vector, $\mathbf{r}$, of an image-plane point relative to the Gaussian image location, when expressed in units of the Rayleigh diffraction scale, $\lambda z_I/R$, in which λ is the imaging wavelength, and δz, z_O, z_I are the distances from the object plane to the in-focus object plane, the in-focus object plane to the pupil plane, and the pupil plane to the image plane, respectively. The symbol $\mathbf{u}$ denotes the position vector in the plane of the pupil, in units of the radius of the pupil. Its polar coordinates are $\mathbf{u} = (u, \phi_u)$. The circular pupil is segmented into L different contiguous annular Fresnel zones, with each zone carrying a spiral phase function, $\psi(\mathbf{u})$, with the number of complete phase cycles changing successively by 1 from one zone to the next. The RPSF can be shown [9] to continuously rotate within the scaled defocus range, $\zeta \in [-\pi L, \pi L]$, as it begins to spread out, break apart, and lose its shape unacceptably outside this range. An illustration of the images of a single point source at different values of the depth parameter ζ, when the RPSF is used, is presented in Fig 2.

With the above formulation, for N point sources the observed image data count $G(x, y)$ at

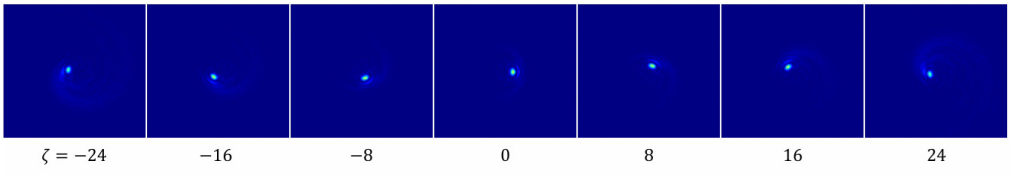


Fig. 2. Single-lobe RPSF images of a single point source for different values of its depth parameter, ζ, for a fixed (x, y) location. An anti-clockwise off-center image rotation about the (x, y) location with increasing ζ is evident.

location (x, y) may be expressed as:

$$G(x,y) \approx \mathcal{N}\left(\sum_{i=1}^{N} A_i(x - x_i, y - y_i) f_i + b\right),$$

where (x_i, y_i) and f_i are the transverse coordinates and flux of the ith point source. Its depth coordinate, z_i, is embedded, via the depth-parameter value ζ_i, in the PSF A_i, b is a uniform background count at each pixel, and $\mathcal{N}$ is the operator for incorporating noise.

Specifically, the forward model for the Gaussian noise case can be conceptualized as following the Gaussian distribution at the p-th pixel,

$$G_p \sim \mathbb{N}([\mathcal{T}(\mathcal{A} * \mathcal{X})]_p + b, \sigma^2), \quad p = 1, 2, ..., d, \tag{1}$$

where $\mathbb{N}(\mu, \sigma^2)$ denotes the Gaussian distribution with expectation μ and variance σ^2, $\mathcal{A} * \mathcal{X}$ is the 3D convolution of $\mathcal{A}$ with $\mathcal{X}$, and $\mathcal{T}$ projects out a 2D slice of the convolution. The symbol $\mathcal{A}$ denotes the 3D PSF dictionary represented as a cube, which is built from a series of images, each corresponding to a different depth, while $\mathcal{X}$ contains the 3D coordinates of the point sources where each entry's value is the corresponding source flux. The total number of pixels in the 2D image array is $d = H \times W$. The forward model for the case of Poisson noise takes a similar form:

$$G_p \sim \mathbb{P}([\mathcal{T}(\mathcal{A} * \mathcal{X})]_p + b), \quad p = 1, 2, ..., d, \tag{2}$$

where $\mathbb{P}(\lambda)$ denotes the Poisson distribution with expectation λ.

2.2. Optimization approach: the variational models

In order to recover the 3D tensor $\mathcal{X}$ from the given observed image G, the optimization approach can be formulated as a minimization problem,

$$\min_{\mathcal{X}} \mathcal{D}\left(\mathcal{T}\left(\mathcal{A} * \mathcal{X}\right) + b, G\right) + \mathcal{R}\left(\mathcal{X}\right),$$

where $\mathcal{D}$ enforces data fitting and $\mathcal{R}(\mathcal{X})$ is an appropriate regularization term. We next formulate these two terms for our two different noise models.

2.2.1. The case of Gaussian noise

Gaussian noise is a common type of noise for which the random error, as we have just noted, has a Gaussian probability distribution. This noise is present due to various factors such as sensor read-out error, non-uniform brightness response of the image sensor, noise and mutual interference from circuit components, and prolonged usage of the image sensor at high temperatures. For the Gaussian noise case, the data-fitting term is a simple quadratic function,

$$\mathcal{D}(\mathcal{T}(\mathcal{A} * \mathcal{X}) + b, G) := \|\mathcal{T}(\mathcal{A} * \mathcal{X}) + b - G\|_F^2,$$

where $\|\cdot\|_F$ denotes the Frobenius norm, namely the ℓ_2 norm, of the vectorized input. The regularization term enforces sparsity, for which we used a non-convex term approaching the ℓ_0 norm for linear least squares data fitting problems. Specifically, we have used the Continuous Exact ℓ_0 (CEL0) penalty function [22] defined as:

$$\mathcal{R}(\mathcal{X}) := \phi_{\text{CEL0}}(\mathcal{X}) = \sum_{i,j,k=1} \phi(\|\mathcal{T}(\mathcal{A} * \delta_{ijk})\|, \mu; \mathcal{X}_{ijk}), \tag{3}$$

where $\phi(a, \mu; u) = \mu - \frac{a^2}{2}\left(|u| - \frac{\sqrt{2\mu}}{a}\right)^2 \mathbb{1}_{\{|u| \le \frac{\sqrt{2\mu}}{a}\}}$ and $\mathbb{1}_E := \begin{cases} 1 & \text{if } u \in E; \\ 0 & \text{others.} \end{cases}$. In addition, δ_{ijk} is a 3D tensor whose only nonzero entry is at (i, j, k) with value 1; μ is the parameter to control the non-convexity. The minimization problem was formulated as

$$\min_{\mathcal{X} \ge 0} \left\{ \|\mathcal{T}(\mathcal{A} * \mathcal{X}) + b - G\|_F^2 + \sum_{i,j,k=1} \phi(\|\mathcal{T}(\mathcal{A} * \delta_{ijk})\|, \mu; \mathcal{X}_{ijk}) \right\}.$$

2.2.2. The case of Poisson noise

The Poisson noise model describes the probability distribution of the number of random events, such as photon counts, occurring per unit time. The data fitting term for the Poisson noise case is the I-divergence, which is also known as the Kullback-Leibler (KL) divergence [23]:

$$\mathcal{D}(\mathcal{T}(\mathcal{A} * \mathcal{X}) + b, g) := D_{KL}(\mathcal{T}(\mathcal{A} * \mathcal{X}) + b, G),$$

where $D_{KL}(z, g) = \langle g, \ln \frac{g}{z} \rangle + \langle 1, z - g \rangle$. The sparsity-enforcing regularization term is designed as a non-convex function [24–26]:

$$\mathcal{R}(\mathcal{X}) := \mu \sum_{i,j,k=1} \theta(a; \mathcal{X}_{ijk}) = \mu \sum_{i,j,k=1} \frac{|\mathcal{X}_{ijk}|}{a + |\mathcal{X}_{ijk}|},$$

where a is a fixed parameter that determines the degree of non-convexity. The Poisson minimization problem was formulated as

$$\min_{\mathcal{X} \ge 0} \left\{ (1, \mathcal{T}(\mathcal{A} * \mathcal{X}) - G \ln(\mathcal{T}(\mathcal{A} * \mathcal{X}) + b)) + \mu \sum_{i,j,k=1} \frac{|\mathcal{X}_{ijk}|}{a + |\mathcal{X}_{ijk}|} \right\}.$$

3. The PINN Methodology

Here we propose a physics-informed neural network called PiLocNet that works for RPSF imaging for the Gaussian and Poisson noise models. As an enhancement of the typical black-box type of neural networks, the proposed model builds the known physics information of the forward process into a PINN framework through additional loss functions.

3.1. Convolutional Neural Network: LocNet

LocNet [15], which combines a deep convolutional neural network (CNN) with a post-processing step, was adopted for RPSF-image based 3D source localization. The CNN part, similar to DeepSTORM3D, consists of 3D grid layers to accommodate the point source prediction. Several practical CNN techniques were employed, such as up-sampling and residual layers. The loss function of LocNet is

$$\mathcal{L}_{\text{LocNet}} = \|\mathcal{G}_{3D} * (\hat{\mathcal{X}} - \mathcal{X}_{\text{GT}})\|_F^2,$$

which is the mean square error of the ground truth $\mathcal{X}_{\rm GT}$ and prediction $\hat{\mathcal{X}}$, with both smoothed by a 3D Gaussian kernel $\mathcal{G}_{\rm 3D}$. After the network generates the initial predictions, a post-processing step further refines the results by treating each cluster of closely spaced point sources as a single source and removing sources with brightness lower than a threshold.

3.2. The pipeline of PiLocNet and its architecture

LocNet is a data-driven approach that only considered the Poisson noise scenario in Ref. [15]. Here we incorporate PINN into LocNet and propose a new framework, PiLocNet, for both Poisson and Gaussian noise cases. The main idea of PINN is to include information about the physics of the problem at hand into the neural network's loss function, as illustrated in the Fig. 1. This approach helps the training process to achieve more accurate results by minimizing the loss with improved guidance provided by known physical information. In this context, we discuss how this concept can be implemented to solve the PSF problem. We modify the LocNet loss function by adding to it two extra terms,

$$\mathcal{L}_{\rm PiLocNet} = w_1 \mathcal{D}(\mathcal{T}(\mathcal{A} * \hat{\mathcal{X}}) + b, G) + \ w_2 \mathcal{R}(\hat{\mathcal{X}}) + w_3 \mathcal{L}_{\rm LocNet}, \tag{4}$$

the first term being the data-fitting term, which contains the PSF operator, $\mathcal{A}$, the known physics information to guide the neural network. However, to add the data fitting term into the loss function correctly, it needs to be model-dependent for different noise types and must, furthermore, be accompanied by an appropriate regularization, which is the second term. Additionally, we need different relative weights, $w_1 : w_2 : w_3$, for the three terms to ensure proper balance and trade-off of these terms.

We employ the same data fitting and regularization terms for the cases of Gaussian and Poisson noise that we described in the previous section, but now allow them to have different relative weights. In other words, we use the following PINN loss functions for the two noise cases, respectively:

$$\mathcal{L}_g = w_1 \|\mathcal{T}(\mathcal{A} * \hat{\mathcal{X}}) + b - G\|_F^2 + w_2 \Phi_{\rm CEL0}(\hat{\mathcal{X}}) + w_3 \|\mathcal{G}_{\rm 3D} * (\hat{\mathcal{X}} - \mathcal{X}_{\rm GT})\|_F^2. \tag{5}$$

and

$$\mathcal{L}_p = w_1 \left\langle 1, \mathcal{T}(\mathcal{A} * \hat{\mathcal{X}}) - G \ln(\mathcal{T}(\mathcal{A} * \hat{\mathcal{X}}) + b) \right\rangle + w_2 \sum_{ijk} \frac{|\hat{\mathcal{X}}_{ijk}|}{|\hat{\mathcal{X}}_{ijk}| + a} + w_3 \|\mathcal{G}_{\rm 3D} * (\hat{\mathcal{X}} - \mathcal{X}_{\rm GT})\|_F^2. \tag{6}$$

The entire loss function, expressed as a weighted summation, facilitates the training process of the neural network via gradient descent. Each component of the loss is computed to yield distinct absolute values, particularly of varying magnitudes. Therefore, the weighting of each loss component is critical in guiding the correct convergence of the neural network. If the loss weights are not appropriately balanced, training might be predominantly influenced by a single term, resulting in undesired outcomes, or potentially causing the network training to fail. For most of the experiments conducted in the paper, the weights w_1, w_2, w_3, were in the ratio 1:700:1000 for the case (5) of Gaussian noise and 1:1:500 for the case (6) of Poisson noise. For experiments in ablation study on different noise levels, one can go through a searching process to reach optimal weight values correspondingly. The approach for the search strategy will be elaborated upon in detail in Section 4.3.

The architecture of PiLocNet closely resembles that of LocNet [17], which was shown to be robust. In a hypothetical experiment with no noise added, the network can output results with very high accuracy, proving that it is well-designed. The network leverages convolutional kernels of specific dimensions to extract pertinent features, followed by batch normalization to expedite convergence. Subsequently, the ReLU activation function is applied. Finally, a shortcut

connection is utilized to merge the input content with the output layer, facilitating residual convolution via a summation layer. The final prediction layer consists of a convolution layer with a convolution kernel size of π and an activation layer with an activation function with a HardTanh range of π. The final output is a 3D lattice image, where the value of each vertex reflects the degree of confidence that there is a point source near the point. The higher the value, the more likely a point source is near the lattice point.

From the output of the network, we obtain a tensor $\hat{\mathcal{X}} \in \mathbb{R}^{H\times W\times D}$ as the initial prediction, where H is the pixel size of height, W is the pixel size of width, and D is the pixel size of depth. The value of the tensor at a node is proportional to the probability of the existence of a real source near this node; we name the value of each node as intensity, denoted as $\hat{\mathcal{X}}_{ijk} \in [0, \pi]$. After initial prediction, we employ the same post-processing as in [13] to $\hat{\mathcal{X}}$, which refines the results by clustering nearby points of 2 pixels, and removing points with brightness lower than a threshold set at 5% of the highest value of the tensor. In this way we obtain a set of points $\hat{\mathbf{X}} = \{\hat{\mathrm{x}}_1, ..., \hat{\mathrm{x}}_m\}$, where $\hat{\mathrm{x}}_i \in ((0, H) \times (0, W) \times (0, D)) \subseteq \mathbb{R}^3$ is a 3D location vector for each $i \in \{1, 2, ..., m\}$.

3.3. Network training

The construction of our training dataset proceeds as follows. We generate a corpus comprising 10,000 images, each defined on a 96×96 pixel array. The flux values for the point sources within each image are drawn from a Poisson distribution with a mean of 2000 photon counts. Subsequently, 90% of the images in this dataset are allocated for training purposes, while the remaining 10% are used for validation. Within this cohort of 10,000 images, the number of point sources is randomly distributed following a uniform distribution from 5 to 50. For our test dataset, we introduce varying numbers, also referred to as densities, of point sources, specifically 5, 10, 15, 20, 25, 30, 35, 40, and 45 sources per image, with the aim to assess our model's efficacy across different source densities. Finally, we generate 100 images for each of these densities, thus 900 images in all, for comprehensive testing. We employ the Adam optimization algorithm [27] in conjunction with a mini-batch size 16. The initial learning rate is stipulated at 1×10^{-3}, with a decay factor of 0.5 applied after every three epochs if there is no discernible improvement in the loss. The termination criterion for the training process is either the absence of any improvement in validation loss over 15 epochs or a validation loss lower than 1×10^{-7}.

The network under consideration is relatively compact and cost-efficient, with a total of 0.3 million parameters. The GPU memory consumption is maintained within 10 GB for the batch size of 16. Such a configuration permits the training to be executed on a single GPU card or distributed across multiple GPUs. Typically, effective training of the model requires between 100 to 180 epochs. Employing an A100 GPU decreases the time per epoch to between 30 to 80 seconds, thereby the overall training duration to 30 to 100 minutes. Also tested with lower resources, training on an RTX 2080Ti GPU, each epoch requires more than 2 minutes, resulting in a total training time of approximately 3 to 5 hours. The inference for processing of 100 test images per group requiring approximately 30 seconds in both devices.

4. Results

To evaluate the 3D-localization performance of our proposed method, we employ recall and precision rates. The **recall rate** is defined as the ratio of the total number of predicted true positives to the total number of point sources that should have been identified as positive. The **precision rate** is similarly defined as the ratio of the total number of true positives to the total number of point sources predicted as positive. True positives are identified based on a specified distance threshold between predicted and ground-truth point sources based on [17]. Note that reducing false negatives improves recall, while reducing false positives improves precision.

4.1. Comparison with previous methods

Based on the experimental setup outlined previously, we compare the average recall and precision rates for PiLocNet with those for three different methodologies: the variational methods [13], the original LocNet method [17], and a modified LocNet v2, which has the same loss function as LocNet except that its architecture is changed to be that of PiLocNet. The primary changes we have made were the removal of the up-sampling layer and adjustments made to the dilation rates within the residual convolution layers. The decision to eliminate the up-sampling layer stemmed from our observation that its removal reduces training time substantially without compromising the model's performance. The dilation rates were aligned from $\{1, 2, 5, 9, 17\}$ to $\{1, 2, 4, 8, 16\}$. A comparative analysis between PiLocNet and LocNet v2 is crucial to ascertain the efficacy of our proposed model in terms of PiLocNet's use of a more physically sensible loss function. The chosen noise model is either Gaussian or Poisson, as described by Eq. 1 or Eq. 2, respectively. For the case of Gaussian noise, its standard deviation, σ, is taken to be uniform across the image and equal to a fraction of the value, $I_{\max}$, of the maximum flux at the pixels in an arbitrary observed image. Unless noted otherwise, we chose $\sigma = 0.1 \times I_{\max}$ for our studies on the Gaussian noise model. We chose the background value of $b = 5$ for both noise models.

Table 1. Evaluation results of $\ell_2 - \text{CEL0}$, LocNet, LocNet v2, and PiLocNet for RPSF images with Gaussian noise

	$\ell_2 - \text{CEL0}$ [13]		LocNet [17]		LocNet v2		PiLocNet_g	
Density	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
10	95.80%	79.72%	**96.00%**	92.15%	93.60%	92.60%	93.60%	**92.62%**
15	93.20%	77.68%	**95.60%**	87.40%	93.73%	88.99%	94.27%	**89.55%**
20	89.30%	72.12%	**92.95%**	81.63%	92.00%	85.21%	92.15%	**85.84%**
30	87.20%	58.77%	88.10%	72.56%	88.27%	79.15%	**88.30%**	**80.12%**
40	77.40%	52.87%	84.28%	63.51%	85.12%	72.44%	**85.23%**	**73.06%**
Average	88.58%	68.23%	**91.39%**	79.45%	90.54%	83.68%	90.71%	**84.24%**

Table 2. Evaluation results of KL-NC, LocNet, LocNet v2, PiLocNet for RPSF images with Poisson noise

	KL – NC [13]		LocNet [17]		LocNet v2		PiLocNet_p	
Density	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
10	99.20%	95.00%	98.90%	96.28%	99.20%	98.63%	**99.30%**	**99.17%**
15	98.80%	89.18%	98.87%	95.54%	99.13%	**98.99%**	**99.33%**	**98.99%**
20	97.55%	85.02%	98.00%	94.45%	97.99%	**98.70%**	**98.95%**	97.85%
30	97.30%	79.54%	96.87%	93.97%	97.53%	95.91%	**97.80%**	**96.67%**
40	95.58%	73.64%	95.00%	90.59%	96.20%	93.83%	**96.43%**	**94.17%**
Average	97.69%	84.48%	97.53%	94.17%	98.15%	97.07%	**98.36%**	**97.37%**

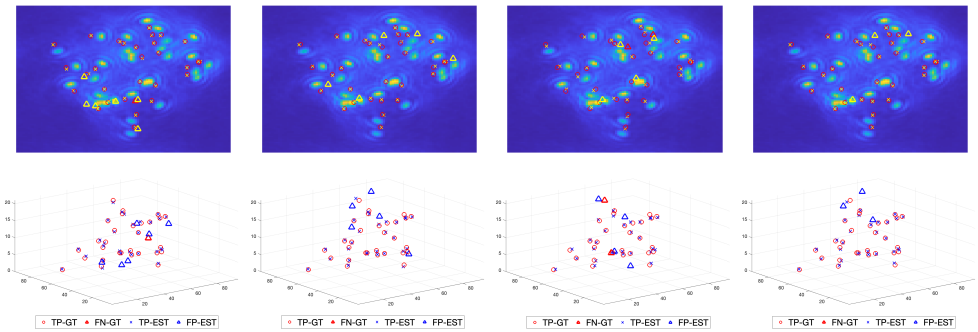
The comprehensive outcomes are presented in Tables 1 and 2, delineating the performance metrics across the aforementioned methods under Gaussian and Poisson noise, respectively. The percentages shown in bold font in each row are the best ones that we obtained for recall and

precision for the corresponding source density for the four methods. We restrict our attention here to only those images in which the number of point sources is either 10, 15, 20, 30, or 40 in order to have a more meaningful comparison with the previously published results of the KL-NC [13] and $\ell_2 - $ CEL0 [13] optimization approaches.

The tabulated results show that both LocNet and PiLocNet, as neural network-based methods, substantially outperform the variational approach in handling images with either Gaussian or Poisson noise. Notably, PiLocNet, with its physics-informed design, shows typically the most impressive results, leading to the highest overall performance metrics for both noise cases. Specifically, PiLocNet improves precision by approximately 0.6% over LocNet v2 in the Gaussian noise scenario and 0.3% in the Poisson noise case. The recall rate has also improved, but not as much as the precision rate. The improvement in precision can be more substantial, however, at higher noise levels, as we will see later in Sec.4.4. This enhancement highlights the efficacy of incorporating physical knowledge into the neural network framework, particularly evident in the precision gains across both types of noise, affirming the value of physics-informed approaches in improving neural network predictions.

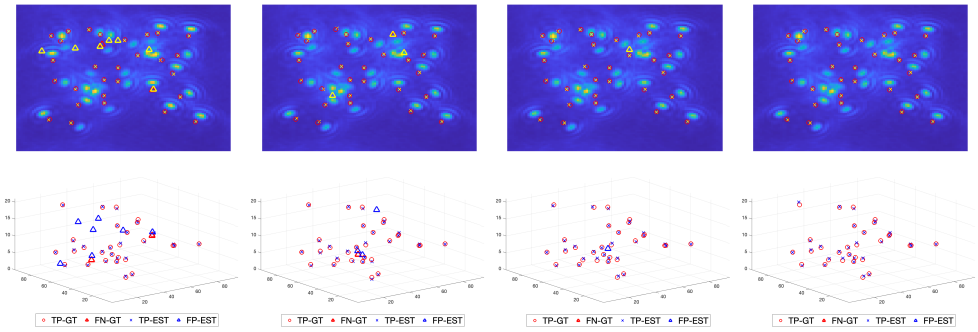
It is noteworthy that LocNet v2 consistently demonstrates a significant improvement in precision rates compared to LocNet [17]. This enhancement stems from LocNet v2's omission of the upsampling algorithm during its operation, which reduces the number of predicted point sources, particularly the false positives.

An example of recovery of sources from their noisy Gaussian and Poisson RPSF image data has been shown in Fig. 3 and Fig. 4. The first row in each figure refers to the same specific 2D snapshot where "o" labels the (x, y) positions of the ground-truth point sources, "x" labels the estimated point sources according to the method used, and "△" represents a mismatch, with the red and yellow colors labeling false-negative and false-positive sources, respectively. The second row shows the locations in 3D grids where the ground-truth point sources are in red markers with red "△" being false-negative and red "o" being true-positive. The estimated source positions are in yellow for the 2D snapshots in the first row and in blue for the 3D grids in the second row, with "△" denoting false-positive and "x" denoting true-positive. It is evident that compared to the original LocNet, LocNet v2 exhibits lower prediction errors. However, it fails at times to predict a ground-truth point source. The fact that when two or more ground-truth point sources are located closely, LocNet v2, having abandoned the upsampling process, is less sensitive to locating such densely packed point sources is the root of such failures. By contrast, since the loss function of PiLocNet incorporates additional information, it more effectively mitigates these errors.



(a) CEL0 [13] (b) LocNet [17] (c) LocNet v2 (d) PiLocNet

Fig. 3. 2D snapshot images (top) and 3D locations (bottom) for the 30-point-source case with Gaussian noise. The triangles denotes the missed matches.



(a) KL-NC [13] (b) LocNet [17] (c) LocNet v2 (d) PiLocNet

Fig. 4. 2D snapshot images (top) and 3D locations (bottom) for the 30-point-source case with Poisson noise. The triangles denote the missed matches (see text for more details).

4.2. *Contributions of the data-fitting and regularization terms*

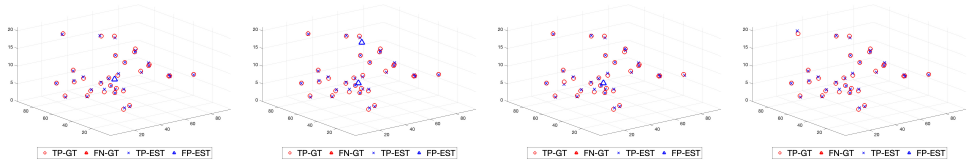
For the purposes of this section, let us rewrite the physics-informed loss function of PiLocNet, namely Eq. 4, in a simplified form, $\mathcal{L}_{\text{PiLocNet}} = w_1\mathcal{D} + w_2\mathcal{R} + w_3\text{MSE}$, where $\mathcal{D}$ and $\mathcal{R}$ are model specific terms. Setting the first two weights to zero, $w_1 = 0, w_2 = 0$, reduces PiLocNet to LocNet v2. For this section, we set up control groups to study the individual contribution of each added term within the loss function, presenting our results in Table 3.

Table 3. Effectiveness of data-fitting and regularization terms for Gaussian and Poisson noised images. The average values of precision and recall among different density cases are shown.

Components			Gaussian noise		Poisson noise	
$\mathcal{D}$	$\mathcal{R}$	MSE	Recall	Precision	Recall	Precision
×	×	✓	90.54%	83.68%	98.15%	97.07%
✓	×	✓	90.67%	82.38%	98.27%	96.33%
×	✓	✓	89.78%	84.19%	97.96%	96.21%
✓	✓	✓	**90.71%**	**84.24%**	**98.36%**	**97.37%**

For the Gaussian case, adding $\mathcal{D}$ to MSE improves the average recall rate from 90.54% to 90.67%, but the average precision drops from 83.68% to 82.38%. When $\mathcal{R}$ is added to MSE, promotes sparsity and thereby r, we increase the precision from 83.68% to 84.19%, while the average recall is not as good as that of the group in which only $\mathcal{D}$ has been added to MSE. For PiLocNet, we find the best average recall and precision rates. Similarly, for the Poisson case, adding $\mathcal{D}$ to MSE improves average recall from 98.15% to 98.27%, the latter being the best recall result among all groups. However, its precision decreases from 97.07% to 96.33%. Combining $\mathcal{D}$, $\mathcal{R}$, and MSE, we once again achieve the best average recall and precision rates.

Fig. 5 illustrates the effect of each added term in the loss function. Simply combining MSE and $\mathcal{D}$ turns some false negatives into true positives, while also introducing some false positives, as evidenced by Fig. 5b, which displays a higher count of falsely estimated points compared to Fig. 5a. In contrast, the regularization term $\mathcal{R}$ tends to elevate the precision rate, as its inclusion within the variational framework acts to control sparsity, thereby mitigating the occurrence of undesired false positives. Fig. 5c is a compelling illustration of how the incorporation of



(a) MSE only. (b) MSE + $\mathcal{D}$. (c) MSE + $\mathcal{R}$. (d) MSE + $\mathcal{D}$ + $\mathcal{R}$.

Fig. 5. The effects of the different components in the loss function Eq. 4. The triangles denotes the missed matches.

regularized terms alone can diminish the occurrence of falsely estimated points. By leveraging both components, PiLocNet balances these effects, leading to an overall enhancement, as seen in Fig. 5d.

4.3. *Optimization of the Relative Weights,* $w_1 : w_2 : w_3$

To correctly incorporate each term and ensure the proper direction of the network's gradient descent, it is crucial to effectively search for the optimal ratio of the weights w_1, w_2, and w_3. The third term, MSE, serves as the foundational loss term, directing the training process and enabling the neural network to align prediction coordinates with the ground truth, thereby offering the most efficient result-driven guidance for network training.

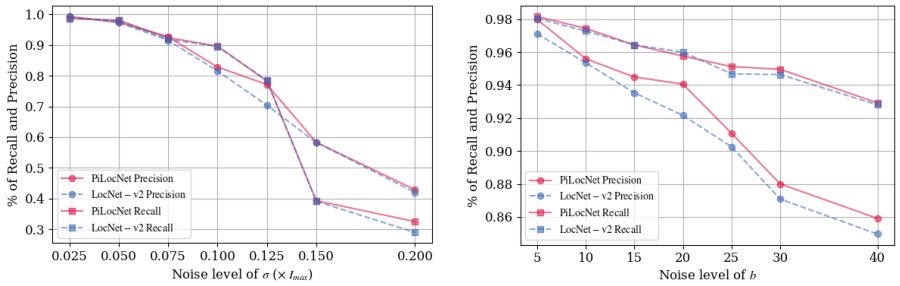
As our results in Sec. 4.2 show, adding the $\mathcal{D}$ term to this MSE term tends to enhance recall because it encapsulates the physical information in the PSF, $\mathcal{A}$, for both noise models. If the weight w_1 is set too low, any improvement in recall might be negligible. Conversely, if w_1 is set too high, the first term could dominate the training. However, since this term simulates a forward process generating the 2D image as compared to the observed image, it can steer the network training only indirectly at best. Thus we still need to rely largely on MSE as the core guiding element in the main training phase. Only an appropriate proportion of the first term $\mathcal{D}$ can enable the network to make more accurate predictions. As previously mentioned and supported by the mathematical model, incorporating only the data-fitting term $\mathcal{D}$ might boost recall, but that may also lead to an increase in false positives, reducing precision. The second term $\mathcal{R}$ is therefore added as a regularization penalty term to mitigate overfitting.

Our search strategy was therefore to start with the third term as the basic foundation for network training, then add the first term, adjust the weight until recall was improved and optimal, and then finally add the second term and adjust its weight to improve and optimize precision. The initial step involved starting with the weight ratio of the first and third terms, $w_1 : w_3$, at five rather disparate values, from 1:1, 1:10, 1:100, 1:1000, to 1:10000, with w_2 set to 0. After finding out the optimal ratio among these five, we checked to see if smaller adjustments of the ratio around it could be made to reach the improve the recall further. A similar subsequent search strategy on the correct $w_2 : w_3$ ratio that optimizes precision without greatly affecting recall allowed us to arrive at the final optimal weight combination, $w_1 : w_2 : w_3$. For this choice of relative weights, we were thus able to improve both recall and precision in the final results obtained by PiLocNet when compared to those obtained by LocNet v2.

4.4. *Robustness on Noise Level*

We also conducted a comprehensive assessment of the robustness of PiLocNet's performance, relative to LocNet v2's, across varying noise levels for the two models of noise considered here. We chose the number of point sources to be 25 for this purpose. In the case of Gaussian noise, we evaluated performance across five different noise levels, as depicted in Fig. 6a, where

the horizontal axis represents the noise level, with the noise standard deviation, σ, being the indicated value multiplied by $I_{\max}$ and the background b fixed at $b = 5$ for each noise level. The results involving Poisson noise are illustrated in Fig. 6b, where the horizontal axis represents the background photon count, b. Across both noise types, PiLocNet consistently outperformed LocNet v2, demonstrating superior precision while maintaining a very similar recall rate (nearly overlapping red and blue □'s). Notably, as the noise level increased, PiLocNet exhibited even more significant precision gains over LocNet v2. For instance, under Gaussian noise with a variance of $\sigma = 0.100 \times I_{\max}$, PiLocNet's precision was higher by about 1%, while at $\sigma = 0.125 \times I_{\max}$, its precision gain exceeded 6%.



(a) Gaussian (b) Poisson

Fig. 6. Precision and recall rates for 25 point sources at 7 different noise levels for the Gaussian and Poisson noise cases.

4.5. *Model Generalizability*

To assess the generalization ability of PiLocNet while training at one noise level and testing at a different noise level, an experiment was conducted to evaluate the network's performance within groups affected by Gaussian noise, categorized by their noise intensity σ ($\times I_{\max}$), as shown in Fig. 7. The Solo-Noise benchmark sets involved training and testing at each of two distinct noise levels of 0.05 and 0.1. The comparison dataset, labeled as Mixed-Noise, is trained with images corrupted with Gaussian noise levels of 0.05 and 0.1, which were evenly represented over the total training volume, which remained at 10 thousands images. During the test phase, the network faced images with a wider range of noise levels, specifically 0.025, 0.05, 0.075, 0.1, and 0.125. Importantly, the images with 0.025, 0.075, and 0.125 noise levels were novel to the network, which were not seen by it during training. The findings indicate that PiLocNet consistently performs well, even for noise levels it had not encountered before during training, thereby demonstrating the model's robustness and its capability to manage noise variations efficiently.

5. Conclusions

The new network, PiLocNet, that we have proposed here adds useful physical information to the neural network by adding to the conventional network's loss function data-fitting and regularization terms that match the noise model governing the observed image data. As we have shown, this greatly improves the network performance. The principal change in the network architecture from LocNet, namely the removal of upsampling and a coarsening of the 3D grid, leads to a reduction of false positives while greatly shortening the network training speed without sacrificing overall performance. PiLocNet outperforms previous methods, as we have demonstrated through robust validation processes.

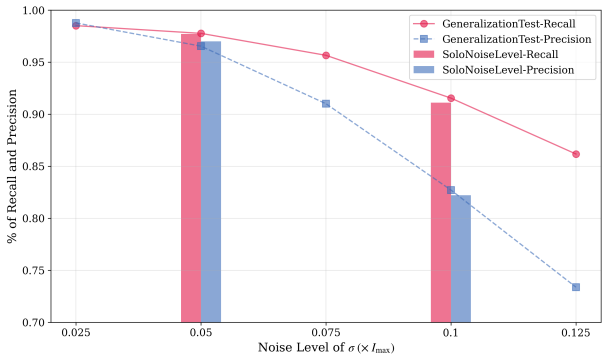


Fig. 7. Generalizability tests under Gaussian noise of PiLocNet, results show the network performs consistently on unseen noise levels. Bar charts: Baseline result of Solo-Noise trained and tested at noise levels 0.05 and 0.1. Line charts: A Mix-Noise group trained with noise level mix (0.05 and 0.1) and tested at 0.025, 0.05, 0.075, 0.1, 0.125.

In the modified loss function of PiLocNet, the data-fitting term $\mathcal{D}$ containing the additional PSF matrix information tends to recover point predictions that would otherwise be missed. The regularization term $\mathcal{R}$, on the other hand, exploits sparsity to reduce the occurrence of false positives. These two effects improve the overall network performance by reducing the rates of both false negatives and false positives.

Neural networks, when well trained, excel at making predictions from highly complex datasets, while variational methods critically employ physical information about the PSF and noise model as well as regularization to avoid overfitting. PiLocNet combines the strengths of both these approaches. By embedding the forward model directly into a neural network, PiLocNet can implement a broad range of PSFs and imaging challenges when the forward model is accurately known.

In future projects, we plan to explore the application of the Physics-Informed Neural Networks (PINNs) methodology across a range of distinct network architectures. Our preliminary results have demonstrated the potential effectiveness of the PINN approach within Vision Transformers (ViT). A potential improvement involves integrating the post-processing steps within the network itself, thus enabling the network to directly produce the final result, rather than the current separate post-hoc stage. Moreover, we plan to broaden our methodology's scope, transitioning from synthetic simulations to include real-world datasets. In our initial evaluations, we identified several obstacles associated with practical imaging, such as color bleaching, flux normalization, magnitude calibration, the size and quality of datasets, and the process of obtaining ground truth labels. Despite these challenges, the application of the PINN approach to practical images remains relevant. Integrating the established physical Point Spread Function (PSF) modeling into the neural network should be achievable, thereby enhancing its learning capabilities and overall performance.

Funding. HKRGC Grants Nos. N_CityU214/19, CityU11301120, C1013-21GF, and CityU11309922; the Natural Science Foundation of China No. 12201286; Guangdong Basic and Applied Research Foundation 2024A1515012347; Shenzhen Science and Technology Program 20231115165836001; CityU Grant 9380101.

Disclosures. The authors declare no conflicts of interest.

Data Availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

References

1. M. K. Ismail, N. Ivan Robert, and H. Ghassan, "A review of super-resolution single-molecule localization microscopy cluster analysis and quantification methods," Patterns **1**, 100038 (2020).
2. R. J. Marsh, K. Pfisterer, P. Bennett, *et al.*, "Artifact-free high-density localization microscopy analysis," Nat. Methods **15**, 689–692 (2018).
3. J. Min, C. Vonesch, H. Kirshner, *et al.*, "FALCON: fast and unbiased reconstruction of high-density super-resolution microscopy data," Sci. Reports **4**, 1–9 (2014).
4. N. Boyd, E. Jonas, H. P. Babcock, and B. Recht, "DeepLoco: Fast 3D localization microscopy using neural networks," BioRxiv p. 267096 (2018).
5. K. Kikuchi, L. D. Adair, J. Lin, *et al.*, "Photochemical mechanisms of fluorophores employed in single-molecule localization microscopy," Angewandte Chemie Int. Ed. **62**, e202204745 (2023).
6. B. Huang, W. Wang, M. Bates, and X. Zhuang, "Three-dimensional super-resolution imaging by stochastic optical reconstruction microscopy," Science **319**, 810–813 (2008).
7. S. R. P. Pavani, M. A. Thompson, J. S. Biteen, *et al.*, "Three-dimensional, single-molecule fluorescence imaging beyond the diffraction limit by using a double-helix point spread function," Proc. Natl. Acad. Sci. **106**, 2995–2999 (2009).
8. Y. Shechtman, L. E. Weiss, A. S. Backer, *et al.*, "Precise three-dimensional scan-free multiple-particle tracking over large axial ranges with tetrapod point spread functions," Nano Lett. **15**, 4194–4199 (2015).
9. S. Prasad, "Rotating point spread function via pupil-phase engineering," Opt. Lett. **38**, 585–587 (2013).
10. R. Kumar and S. Prasad, "PSF rotation with changing defocus and applications to 3D imaging for space situational awareness," in *Proc. AMOS Tech. Conf., Maui, HI,* (2013).
11. T. Teuber, G. Steidl, and R. H. Chan, "Minimization and parameter estimation for seminorm regularization models with I-divergence constraints," Inverse Probl. **29**, 035007 (2013).
12. C. Wang, G. Ballard, R. Plemmons, and S. Prasad, "Joint 3D localization and classification of space debris using a multispectral rotating point spread function," Appl. Opt. **58**, 8598–8611 (2019).
13. C. Wang, R. H. Chan, R. J. Plemmons, and S. Prasad, "Point spread function engineering for 3D imaging of space debris using a continuous exact ℓ_0 penalty (CEL0) based algorithm," in *Int. W. Imag. Proces. & Inverse Probl.*, (Springer, 2018), pp. 1–12.
14. C. Wang, R. Chan, M. Nikolova, *et al.*, "Nonconvex optimization for 3-dimensional point source localization using a rotating point spread function," SIAM J. on Imaging Sci. **12**, 259–286 (2019).
15. E. Nehme, D. Freedman, R. Gordon, *et al.*, "DeepSTORM3D : dense 3D localization microscopy and PSF design by deep learning," Nat. Methods **17**, 734–740 (2020).
16. A. Speiser, L.-R. Mller, P. Hoess, *et al.*, "Deep learning enables fast and dense single-molecule localization with high accuracy," Nat. Methods **18**, 1082–1090 (2021).
17. L. Dai, M. Lu, C. Wang, *et al.*, "LocNet: deep learning-based localization on a rotating point spread function with applications to telescope imaging," Opt. Express **31**, 39341–39355 (2023).
18. M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," J. Of Comput. Phys. **378**, 686–707 (2019).
19. E. Xypakis, V. deTurris1, F. Gala, *et al.*, "Physics informed deep learning for microscopy," in *EPJ Web Conf.*, (2022), p. 04007.
20. K. Wang and E. Y. Lam, "Deep learning phase recovery: data-driven, physics-driven, or a combination of both?" Adv. Photonics Nexus **3**, 056006–056006 (2024).
21. A. Tsakyridis, M. Moralis-Pegios, G. Giamougiannis, *et al.*, "Photonic neural networks and optics-informed deep learning fundamentals," APL Photonics **9** (2024).
22. E. Soubies, L. Blanc-Féraud, and G. Aubert, "A continuous exact ℓ_0 penalty (CEL0) for least squares regularized problem," SIAM J. Imag. Sci. **8**, 1607–1639 (2015).
23. T. Le, R. Chartrand, and T. Asaki, "A variational approach to reconstructing images corrupted by poisson noise," J. Of Math. Imaging And Vis. **27**, 257–263 (2007).
24. M. Nikolova, M. K. Ng, S. Zhang, and W.-K. Ching, "Efficient reconstruction of piecewise constant images using nonsmooth nonconvex minimization," SIAM J. Sci. Comput. **1**, 2–25 (2008).
25. M. Nikolova, M. K. Ng, and C.-P. Tam, "On ℓ_1 data fitting and concave regularization for image recovery," SIAM J. Sci. Comput. **35**, A397–A430 (2013).
26. J. Xiao, M. K.-P. Ng, and Y.-F. Yang, "On the convergence of nonconvex minimization methods for image recovery," IEEE Trans. Image Process. **24**, 1587–1598 (2015).
27. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR,* (2015).